Discover Artificial Intelligence

RESEARCH Open Access

Adaptive object detection in dynamic road environments using YOLOv11 with continuous learning for Real-Time obstacle detection

Nkalubo Lenard Byenkya[1*], Nakibuule Rose[2] and Okila Nixson[3]

*Correspondence:
Nkalubo Lenard Byenkya
lenkalubo@gmail.com
[1]Department of Networks, Data Science and Artificial Intelligence, Kyambogo University, Kampala, Uganda
[2]Department of Computer Science, Makerere University, Kampala, Uganda
[3]Department of Computer Science, Lira University, Lira, Uganda

Abstract

Real-time object detection in dynamic road environments is crucial for enhancing road safety, infrastructure monitoring, and Intelligent Transportation Systems (ITS). This study presents an adaptive object detection model based on YOLOv11 with continuous learning, designed to detect road obstacles in real-time without frequent retraining autonomously. Unlike existing YOLO-based models such as YOLOv10 and LD-YOLOv10, the proposed model integrates a continuous learning mechanism, enabling it to adapt dynamically to evolving road conditions, including potholes, vehicles, signposts, and other obstacles. The model was trained and evaluated on a dataset collected from diverse urban road environments in Uganda, achieving a mean average precision (mAP) of 0.856, precision of 0.873, and recall of 0.849. It also demonstrates high-speed performance at 78.7 Frames Per Second (FPS), making it suitable for real-time deployment on resource-constrained edge devices. While this study primarily focuses on model development and optimization for road obstacle detection, future research will explore its potential application in assistive navigation systems for visually impaired individuals. Comparative analysis against state-of-the-art models, including YOLOv10, LD-YOLOv10, and Tiny-DSOD, demonstrates the competitive performance of the proposed model, particularly in detecting small and occluded objects.

Keywords Adaptive object detection, Continuous learning, YOLOv11, Real-Time road monitoring, Obstacle detection

1 Introduction

Object detection in dynamic road environments is essential for real-time monitoring, road safety enhancement, and Intelligent Transportation Systems (ITS) [1]. Roads are constantly affected by evolving environmental conditions, changing traffic patterns, and emerging obstacles such as potholes, vehicles, road signs, and pedestrian pathways [2]. However, traditional object detection models struggle to adapt to these variations because they rely on static datasets that do not account for real-world changes [3].

Recent advancements in deep learning-based object detection have significantly improved real-time applications in road monitoring. Variations of You Only Look Once

(YOLO), such as YOLOv8 and YOLOv10, have demonstrated high performance in low-light conditions and resource-efficient deployments [4–6]. However, these models lack adaptive learning mechanisms, making them less effective in environments where road conditions frequently change. For example, YOLOv10 has demonstrated strong accuracy in vehicle detection but does not support continuous learning to accommodate newly appearing obstacles [6, 7]. Likewise, LD-YOLOv10 [8], optimized for lightweight drone deployment, lacks the capability for real-time model adaptation, limiting its applicability in long-term road infrastructure monitoring [8]. These challenges highlight the need for a continuous learning model that dynamically updates its knowledge during real-world deployment.

While transformer-based object detection models offer high accuracy, their computational complexity makes them unsuitable for edge deployment in resource-constrained environments [6, 9]. Moreover, most existing road monitoring systems rely on extensive preprocessing techniques and manual retraining, reducing their adaptability and scalability for real-time applications [10, 11]. This study addresses these challenges by proposing an optimized YOLOv11 model integrated with continuous learning, designed to enhance real-time obstacle detection in dynamic road environments.

Unlike previous models, our proposed approach supports real-time learning without requiring full retraining, making it more effective in detecting newly emerging road obstacles. The model leverages multi-scale feature representation to enhance the detection of small and irregularly shaped objects such as potholes. This feature improves the model's ability to capture object variations under diverse environmental conditions, including occlusions and lighting changes. The proposed model is highly adaptable and suitable for deployment in low-resource environments where frequent retraining is infeasible. To ensure real-world adaptability, the continuous learning framework allows the model to dynamically adjust to evolving road conditions without human intervention. Unlike conventional object detection models trained on static datasets, our approach learns from newly encountered obstacles, environmental variations (e.g., lighting and weather changes), and road modifications in real-time. The primary focus of this study is on developing and fine-tuning an adaptive object detection model for real-time obstacle detection in dynamic road environments. While the model has potential applications in assistive navigation for visually impaired individuals, this remains a future research direction beyond the scope of this study. Future work will explore how the model can be extended to assist visually impaired pedestrians by detecting sidewalks, stairs, and moving obstacles, supporting AI-driven assistive mobility solutions. The study was guided by the following research question:

- *How does the integrated continuous learning with cross-scale feature fusion into the optimized YOLOv11 model contribute to an increase in the detection accuracy compared to baseline models?*

The rest of this paper is structured as follows: Sect. 2 reviews related literature on deep learning-based object detection and continuous learning. Section 3 details the dataset, preprocessing techniques, and YOLOv11 model architecture. Section 4 presents experimental results and performance evaluations, followed by Sect. 5, which discusses the broader implications of the findings. Section 6 concludes the study and outlines future research directions.

1.1 Motivation of the study

Roads are of utmost importance to economic development; in developing countries like Uganda, transportation is very important for trade, agriculture, and daily livelihood. However, the roads in Uganda are always deteriorating because of high-intensity traffic volumes, poor maintenance, and environmental conditions that eventually lead to potholes [2]. These potholes are a significant safety hazard, contribute to increased transportation costs, and lead to road accidents. Despite the existence of government efforts to maintain roads, manual pothole detection is inefficient, resource-intensive, and costly [12]. Recent breakthroughs in deep learning and computer vision have provided tools for automatic object detection, with YOLO-based models emerging as state-of-the-art methods due to their speed and accuracy [13]. However, none of the existing variants of YOLO, such as YOLOv10 and LD-YOLOv10, have developed any adaptive mechanisms for dynamic environments, including rapidly changing road conditions, lighting variations, and occlusions [5, 8]. Most of these models are designed based on fixed datasets and architectures, limiting their scalability and performance in real-world applications [9]. Moreover, although bounding box-based object detection is a core operation in computer vision, continuous learning for refining and improving the detection over time is not considered in many models [14].

The basis of this work, therefore, is motivated by the need to respond to these challenges and contribute further to the field of computer vision with an optimized YOLOv11 model with continuous learning. Unlike other object detection models, this study will take a different approach in which continuous learning is integrated into the model to adapt to new road conditions without frequent retraining for efficient, more accurate, and scalable pothole detection in real-world deployments within Uganda's road monitoring systems. This work's key novelty consists of optimizing not only the YOLOv11 but also testing it in more realistic application contexts. To that end, this model has been trained and tested on a dataset collected in Uganda, representing multiple scenarios of weather, lighting conditions, and roads. Results of the research are obtained through bounding box prediction, one of the most fundamental features of computer vision. These give concrete evidence that the model is indeed capable of detecting potholes, vehicles, walkways, and signposts. The bounding box results show the practical applicability of the study; hence, it shows AI research output. This work bridges the gap between the theoretical optimization of object detection models and real applications by optimizing YOLOv11 with continuous learning toward real-time detection of potholes in roads. The innovation targets improving road safety, reducing maintenance costs, and enhancing transportation efficiency in Uganda, while in the future being deployable on Internet of Things (IoT) edge devices to reach scalable and cost-effective smart road monitoring systems.

2 Background

2.1 Fundamental concepts and algorithms

(a) YOLO11.

The You Only Look Once (YOLO) object detection algorithm, first put forward in the year 2016, transformed real-time object detection by encapsulating region proposal and classification processes into one single neural network, an approach that markedly reduced computation times [15]. Its application has been immense in autonomous

vehicles and traffic safety to develop safety in road transportation and aviation, object detection in real-time to avoid collisions, traffic monitoring, pedestrian detection, detection of traffic signs, and aviation hazard analysis, among many more. YOLOv1 forms the basis for further development in the succeeding series, up to the current version, YOLOv11, in Fig. 1. YOLOv11 is yet another state-of-the-art model, adding new contributions to its already successful series, with more features and enhancements to improve further speed and flexibility [16]. From YOLOv11, one would expect excellent speed, precision, and simplicity, hence value across a wide span of object detection and tracking tasks, instance segmentation, image classification, and pose estimation. Compared to other detection networks, YOLOv11 is more adaptable, as it supports more general computer vision tasks than object detection alone [17]. This includes posture estimation and instance segmentation, further extending the model's applications across different areas. The balance of power and practicality has been considered in the design of YOLOv11, which can solve specific challenges of different industries with higher accuracy and efficiency. The development of real-time object detection technology keeps going with this latest model, pushing the boundary of what's possible in applications with computer vision. With its versatility and enhancement in performance, YOLOv11 could be another leap into the future of opening wide new horizons for real-world implementations across diversified sectors.

(b) Continuous Learning for Object Detection.

The main goal of object detection continual learning is to make a model learn a stream of tasks $[t1, t2, t3, ...]$ while successfully localizing and recognizing all the classes of interest over time [18]. As seen in Fig. 2, continual learning for object detection is an evolving field of research that comes with considerable practical value. The methodologies in this domain are mainly divided into Class-Incremental Object Detection (CIOD) and Domain-Incremental Object Detection (DIOD). Most works considered in CIOD deal with models trained with an initial set of base classes, and then subsequently fine-tuned to incrementally recognize new classes not seen so far. These applications are especially relevant in all those scenarios that require adaptation to dynamically changing environments, like road condition monitoring or autonomous navigation, where new types of objects can appear over time. By contrast, the DIOD focuses on the scenario where the object classes are fixed, but their distribution may change dynamically due

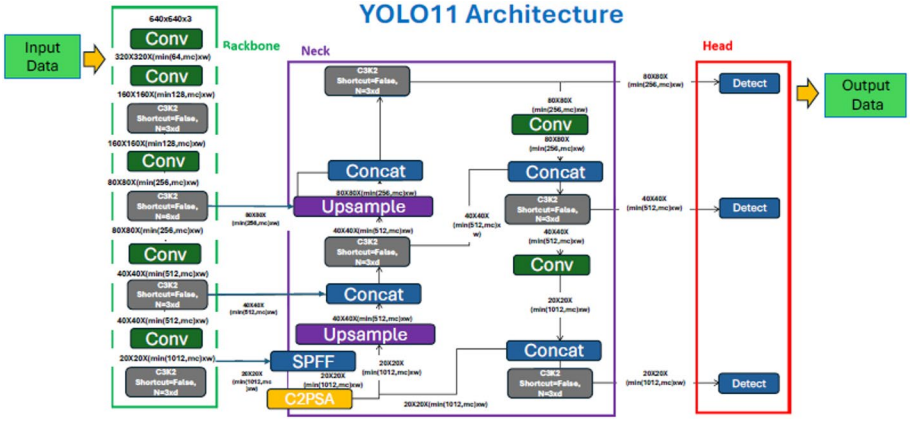

Fig. 1 YOLOv11 architecture diagram; adapted from [17]

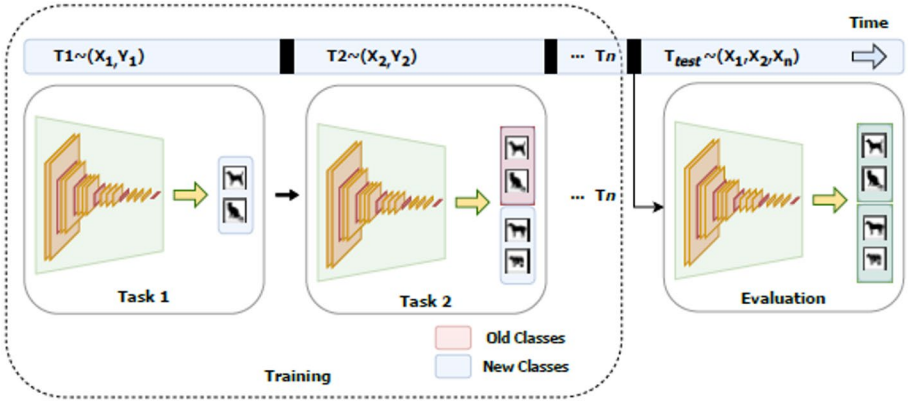

Fig. 2 A generic class-incremental scenario for object detection. Adapted from [18].

to environmental changes. In this case, a model must maintain its ability to correctly detect objects under any shifts in contextual factors, including illumination conditions, weather conditions, or road conditions. Recent advances in DIOD suggest that strategies primarily designed for classification tasks, such as random replay and increasing network capacity, can yield competitive performance even under challenging conditions. However, CIOD poses greater challenges, as the task of incrementally adding new classes to an existing object detector requires specialized mechanisms to mitigate catastrophic forgetting while efficiently managing memory and computational constraints [19]. Considering the dynamic nature of road conditions, with continuously arising obstacles such as potholes, signs, and debris, the study focuses on Class-Incremental Object Detection. The model therefore embeds YOLOv11 with Continual Learning for a system that can adapt to novel road conditions without the need for frequent retraining, hence highly feasible in real time for dynamic, resource-constrained environments. Continuous learning can help in adapting object detection under the continuous variation of object appearances, lighting conditions, and backgrounds [14]. Continual learning empowers the model to learn incrementally from the newly coming data while preserving past knowledge, hence reducing the frequent need for retraining [20]. In this work, YOLOv11 is combined with continuous learning using Class-Incremental Object Detection; this allows the model to adapt to changes in road conditions, such as the appearance of new potholes, road signs, or obstacles that could be highly suitable for real-world deployments in road infrastructure monitoring.

2.2 Adaptive object detection

The study [21] proposed OCC-YOLOX, an enhanced YOLOX algorithm based on adaptive deformable convolution, to increase the accuracy of target detection in challenging situations. The approach uses Fast Spatial Pyramid Pooling, Overlapping IoU loss function, and coordinated attention to maximize bounding boxes. Experiments demonstrate that OCC-YOLOX improves object detection efficiency and accuracy by outperforming current detection algorithms in complicated occlusion scenarios. In intelligent transportation systems, this paper provides fresh perspectives on obscured target detection.

The study [22] Domain Adaptive Object Detection for Remote Sensing Images addresses the challenge of RS data privacy and transmission difficulties. The method

Table 1 Related literature on advances and limitations

Author(s)	Objectives	Results	Limitations
Kamath & Renuka [42]	Review trends in resource-constrained object detection using deep learning.	Identified trends, challenges, and gaps in edge-based detection.	Limited adaptation and continuous learning capabilities.
Nicolas & Nicolas [4]	Develop object detection in low-illumination environments using YOLOv8.	Achieved mAP 71.2% and AP 77.9% using enhanced YOLOv8.	Performance drops without low-light pre-processing.
Mercaldo et al. [9]	Detect road signs for real-time traffic management using YOLO.	Reliable real-time detection of road signs on large datasets.	Not designed for continuous learning or adaptive updates.
Yi & Anantrasirichai [43]	Improve object tracking under noise and low light using transformers.	Improved tracking accuracy in low-light environments.	Requires significant data preprocessing and complex models.
Y. Li et al. [5]	Propose Tiny-DSOD for ultra-efficient object detection.	Outperformed Tiny-YOLO, MobileNet-SSD, and SqueezeDet.	High computational cost despite improved efficiency.
J. Li et al. [44]	Develop YOLO_GD for detection in low light with spatial attention.	5.97% improvement in mAP over YOLOv5.	Limited application beyond low-light settings.
Royer & Lampert [45]	Introduce adaptive detection for variable object sizes and densities.	Accurate and efficient detection using multi-stage processing.	Complex model design for multi-stage processing.
M. Liu et al. [46]	Compress YOLOv5 using Tensor Train decomposition for hardware efficiency.	Reduced model size with hardware-compatible acceleration.	Hardware-specific, limiting general deployment.
Abba et al. [10]	Develop real-time detection for security systems using deep learning.	Accurate security detection with high performance on MOT datasets.	Limited scalability and lack of advanced adaptability.
Wahab et al. [47]	Create real-time object recognition using pre-trained models.	Achieved 97% accuracy using various freely available datasets.	Requires large pre-trained models and frequent retraining.
Lago et al. [49]	Enable lightweight object detection on aerial remote sensing devices.	High detection accuracy with embedded hardware and minimal overhead.	Low generalization for complex, real-world conditions.
Ye et al. [50]	Enhance home robot detection with deblurring and YOLO-like techniques.	Real-time object detection with enhanced small object sensitivity.	Heavy reliance on deblurring and noise reduction methods.
Z. Liu et al. [51]	Develop a memory-efficient UAV detection model using FasterNet-16.	Superior capabilities for UAV-based detection with multi-scale fusion.	Resource-heavy model not suitable for low-resource environments.
Aamir et al. [52]	Improve small object detection in cluttered environments using SSD.	High detection accuracy with improved performance in cluttered settings.	Limited small object detection under occlusions.

comprises perturbed domain generation and alignment, requiring consistent features across the teacher-student network. The method's effectiveness has been validated through synthetic-to-real and cross-sensor experiments, and its application in computer vision datasets is demonstrated. The method can be extended to other fields, allowing for more efficient and accurate detection of domain-variant features.

The study [23] proposes an adaptive object detection method for interactive settings. It aims to detect objects in images observed by an embodied agent in different environments. The model adapts to different appearance characteristics and performs on par with a model trained with full supervision. This approach improves object detection by 11.8 points over a high-performance object detector, DEtection TRansformer (DETR) [24]. The model continues training during inference and adapts to environments with different appearance characteristics.

The study [25] presents an unsupervised domain adaptation framework for object detection that addresses the style and weather gaps in real-world environments. It resolves the style gap by focusing on high-level features' style-related information and reduces the weather gap using self-supervised contrastive learning. Extensive experiments show that this method outperforms other methods for object detection in adverse weather conditions.

The study [26] addresses the domain adaptation problem in object detection, focusing on robust learning. It proposes a robust object detection framework that is resilient to noise in bounding box class labels, locations, and size annotations. The model is trained on the target domain using noisy object bounding boxes obtained from a detection model trained in the source domain.

A novel transformer-based adaptive object detection method is proposed for accurately detecting multi-scale remote sensing objects in complex backgrounds [27]. The method uses a dual attention vision transformer network and an adaptive path aggregation network, with CBAM suppressing background information. Experiments on RSOD, NWPU VHR-10, and DIOR show that the method outperforms compared object detection methods, providing more effective feature information.

This study [28] reviews the progress in Deep Domain Adaptive Object Detection (DDAOD) methods, introducing basic concepts, categorizing detectors into five, providing detailed descriptions of methods, and offering insights for future research trends, highlighting the need for more labeled training data and identical distributions in practice.

The study [29] proposed an adaptive multi-scale feature enhancement and fusion module (ASEM) to improve object detection. The method uses a feature pyramid to gather coarse multi-scale features, then refines them using atrous convolutions and an attention mechanism for feature fusion. The method achieves an mAP of 74.21% on the DOTA-v1.0 dataset and 84.90% on the HRSC2016 dataset, demonstrating its effectiveness and practicality in multi-scale remote sensing object detection tasks.

2.3 Edge AI for real-time object detection

The study [30] explored advanced computer vision techniques for Intelligent Transportation Systems (ITS), specifically on vehicles. It uses Roadside Units (RSUs) to offload DT data, updating edge detection results on the DT. This Distributed Edge Intelligence (DEI) enables rapid decision-making with reduced latency and provides a comprehensive view of vehicle lifecycle management. The proposed algorithm achieves an mAP of 85% using YOLOv9t with minimal latency, ensuring effective pothole detection and seamless communication among nearby vehicles.

The [31] study explores the potential of deep learning models for pothole detection on asphalt pavements. Three models, Tiny-YOLOv4, YOLOv4, and YOLOv5, were deployed on an AI kit (OAK-D) on a Raspberry Pi for real-time detection. The models achieved a high mean average precision (mAP) of 80.04%, 85.48%, and 95% on an image dataset with potholes, proving their strength and suitability for real-time pothole detection.

Defects in the road surface are a big problem for traffic movement, leading to dangers and accidents. Autonomous cars can assist in identifying and fixing these flaws by utilizing deep learning methods and in-car sensors. To solve this problem, the

Intelligent Transportation Systems (ITS) developed the idea of a vehicular ad hoc network (VANET). To provide future vehicles with road information, a unique system for autonomous vehicles to automatically detect road irregularities is presented [32]. Road abnormalities are classified using methods like Visual Geometry Group and Residual Convolutional Neural Network, which lowers accident rates and dangers.

The study [33] evaluated object detection models like YOLOv8, EfficientDet Lite, and SSD on edge devices like Raspberry Pi and Jetson Orin Nano. Results show that lower models are more energy-efficient and faster in inference, while higher models consume more energy and have slower inference. Jetson Orin Nano is the fastest and most energy-efficient option for request handling, despite high idle energy consumption. These findings emphasize the need for balancing accuracy, speed, and energy efficiency in deep learning deployment.

The study [34] presents an efficient, low-complexity, and anchor-free object detector using the YOLO framework. It uses enhanced data augmentation to suppress overfitting and a hybrid random loss function for small object detection. The model achieves high accuracy on edge computing devices and is designed for lower computing power.

The study [35] explores deep learning-based lightweight object detection models for edge devices, addressing the growing demand for accurate, quick, and low-latency models. It provides a comprehensive description of recent methods, lightweight backbone architectures, training and inference processes, and various applications. The study examines the performance of conventional deep learning-based lightweight object detectors on datasets like MS-COCO [36] and PASCAL-VOC [37].

The study [38] investigates the deployment of RetinaNet and PointPillars object detection networks on an edge AI platform. The researchers found slight advantages for convolutional layers and TorchScript for fully connected layers. Quantization reduced runtime while maintaining detection performance.

The study [39] investigates a cooperative task execution system in an edge-computing architecture, focusing on optimizing execution accuracy and time for inferencing deep learning models. The system offloads workloads to end devices, which collectively execute object detection on transmitted images. The authors focus on implementing new policies to optimize E2E execution time and accuracy, highlighting the role of effective image compression and batch sizes. They use YOLOv5 and evaluate the performance of the model using metrics like Precision, Recall, and mAP.

The study [40] proposes a new edge GPU-friendly module for multi-scale feature interaction, addressing the issue of compressing deep neural network-based object detection models to edge devices. The module exploits missing connections between feature scales and uses a transfer learning backbone to improve accuracy and execution speed on edge GPU devices. The module outperforms its predecessors, achieving faster performance on PASCAL-VOC and COCO.

The study [41] proposes an object detection system called Edge YOLO, based on edge-cloud cooperation and reconstructive convolutional neural networks, to address the challenges of low timeliness and high energy consumption in existing DL-OD schemes for autonomous vehicles. The lightweight framework combines pruning feature extraction and compression feature fusion networks, enhancing efficiency and reducing dependence on computing power. The system has been tested on COCO2017 and KITTI datasets, achieving a 47.3% accuracy rate.

2.4 Object detection on road infrastructure monitoring and pothole detection

The study [42] aimed to find the current trends of resource-constrained applications in deep learning-based object detectors. 167 studies were systematically reviewed to understand the problem and techniques used. The conclusion discusses gaps, possibilities, and future perspectives in the field, indicating that it has grown significantly in the last decade. The study further focused on lightweight, efficient deep learning-based detectors for edge devices.

The study [4] presents a deep learning-based model for object detection in low-illumination environments, using YOLOv8 and AdamW optimizer. Pre-processing techniques were employed to enhance images, reduce noise, and emphasize edges. The ExDARK database was used for training. The model achieved a mean average precision of 71.2% and an average precision of 77.9%, surpassing state-of-the-art approaches.

The study [9] proposed a method using deep learning to detect road signs using the You Only Look Once model. Experiments on a 10,000-image dataset show the model's reliability for real-time detection of road signs, crucial for efficient traffic management and autonomous vehicle operation.

This study [43] explored the impact of noise, color imbalance, and low contrast on automatic object trackers in low-light environments. It proposes integrating denoising and low-light enhancement methods into a transformer-based system, resulting in an outperforming tracker.

The study [5] proposed a new object detection framework, Tiny-DSOD, based on the Deeply Supervised Object Detection (DSOD) framework. It introduces two innovative architecture blocks: Depthwise Dense Block (DDB) and Depthwise Feature Pyramid Network (D-FPN). Experiments show that Tiny-DSOD outperforms state-of-the-art ultra-efficient solutions like Tiny-YOLO, MobileNet-SSD, SqueezeDet, and Pelee in all metrics, including parameter size, FLOPs, and accuracy.

The study [44] presents an efficient object detection network, YOLO_GD, designed for precise detection in low-light environments. It uses a cross-layer feature fusion method, a Bi-level routing spatial attention module, and a probabilistic distance metric to improve detection accuracy. Experimental results show a 5.97% improvement in mean average precision compared to YOLOv5, enhancing object detection performance in low-light conditions. The network's generalization capability is further enhanced in occluded scenarios.

The study [45] proposes a flexible detection scheme that adapts to variable object sizes and densities. It uses a sequence of detection stages that predict groups of objects and individuals, sparing computational effort by discarding irrelevant regions. This method also allows working with lower image resolutions in the early stages, where groups are more easily detected. Experimental results show that the proposed method is as accurate and computationally more efficient than standard single-shot detectors across three different backbone architectures.

The study [46] proposed an efficient real-time object detection framework on resource-constrained hardware devices using Tensor Train decomposition. The method compresses the YOLOv5 model, reducing model size and improving execution time, based on FPGA devices, and combines hardware and software co-design.

The study [10] presents a real-time framework for object detection and tracking for security surveillance systems, based on approximate median filtering, component

labeling, background subtraction, and deep learning approaches. The algorithms are implemented using Python and C# programming languages, and the framework is evaluated for accuracy and precision on MOT15, MOT16, and MOT17 datasets. The study revealed that future studies will consider dynamic scalability, multiple data sources, and object-detection techniques like You Only Look Once (YOLO) to adapt to complex situations.

This study [47] investigates real-time object detection and recognition systems using computer vision and human-computer interaction. The researchers use the single-shot detector algorithm and deep learning techniques with pre-trained models to detect static and moving objects in real-time. They evaluate pre-trained models on various datasets and propose a highly accurate system with a 97% accuracy rate. The system is operational on reasonable equipment and utilizes freely available datasets like MS COCO [36], PASCAL-VOC [37], and KITTI [48].

The study [49] presents a lightweight, low-cost, and modular approach to real-time object detection and tracking in aerial remote sensing. It integrates real-time object detection models with affordable embedded hardware, allowing minimal computational overhead and real-time applications. The system generates cleaner areas of interest, with real-time processing speeds and GPS positioning accuracy within a meter.

The study [50] revealed that home service robots require real-time object detection for efficient service tasks. However, blurred images in motion pose challenges, especially for small objects like fruits and tableware. A dynamic and real-time object detection algorithm was proposed, including an image deblurring and object detection algorithm. The DA-Multi-DCGAN algorithm improves image clarity and Structural Similarity, while the AT-LI-YOLO method enhances small and occluded object detection. The method uses depthwise separable convolution and a lightweight network unit, Lightblock, improving sensitivity and precision. The proposed algorithm requires a 29 ms processing time, meeting real-time vision requirements.

The study [51] presents a lightweight, memory-efficient model for real-time object detection in unmanned aerial vehicles (UAVs). It incorporates the FasterNet-16 backbone and a novel multi-scale feature-fusion technique. The model's performance evaluations show superior capabilities and robustness in real-time operations, advancing UAV deployment in remote-sensing tasks and improving productivity and safety.

The study [52] focuses on improving the detection accuracy of small, undetermined moving objects in occluded environments with cluttered backgrounds using SSD and YOLO algorithms. The method uses a custom dataset, preprocessing techniques, and data augmentation techniques to improve precision. The SSD-Mobilenetv2 model outperforms YOLOv3 and YOLOv4, with higher accuracy and frame per second. Techniques like data enhancement, noise reduction, parameter optimization, and model fusion enhance detection effectiveness. The method also includes a graphical user interface system.

The study [8] proposes a lightweight drone target detection algorithm, LD-YOLOv10, to improve detection performance on edge devices. The algorithm uses a lightweight feature extraction structure called RGELAN, an AIFI module, a DR-PAN Neck structure, and bounding box regression loss functions. Experiments show that LD-YOLOv10 reduces parameter number by 62.4%, increases accuracy slightly, and has faster detection

speed. When deployed on the low-power NVIDIA Jetson Orin Nano device, it achieves a detection speed of 25 FPS.

The study [6] compares two deep-learning models, YOLOv8 and YOLOv10, for vehicle detection across various classes. Results show that YOLOv10 outperforms YOLOv8, especially in smaller, complex vehicles like bicycles and trucks. YOLOv8 has slight advantages in car detection, but both models perform similarly for buses and motorcycles. The study provides insights into real-world applications and optimizes YOLO architectures for specific vehicle detection tasks.

The study [53] did a survey exploring the potential of various YOLO variants, including YOLOv10, in agricultural advancements. It aims to identify challenges in agriculture, assess YOLO's advancements, and explore its applications. The survey also critically analyzes YOLO's performance and future trends, providing insights into the evolving relationship between YOLO variants and agriculture. The findings contribute to a better understanding of precision farming and sustainable practices.

The review [11] explores the YOLO architecture and its derivatives, focusing on their improvements in quality inspection in the photovoltaic (PV) domain. Key drivers include path aggregation networks, efficient layer aggregation architectures, and programmable gradient information. Future research will focus on refining YOLO variants to tackle a wider range of PV fault scenarios, including attention mechanisms.

2.5 Current object detection advances and limitations

The reviewed studies in Table 1 highlight significant advancements in object detection within low-resource environments, emphasizing efficient, lightweight, and scalable models. Techniques such as YOLO-based architectures, transformer-based trackers, and lightweight frameworks like Tiny-DSOD and FasterNet-16 demonstrated improved detection accuracy, reduced computational overhead, and real-time adaptability. Despite these developments, there are still some major limitations, such as the inability for continuously learning, heavy reliance on specific hardware, susceptibility to changes in environmental conditions like low light, and degradation of performance without preprocessing methods. These issues have brought up the need for adaptive models that can learn dynamically from constantly changing data while maintaining efficiency in resource-constrained environments, hence motivating this work of continuous learning integrated with YOLOv10 for adaptive object detection.

3 Materials and methods

3.1 The study area

The dataset was collected in Kampala, Uganda, in Africa, shown in Fig. 3. The images were collected from all 5 divisions of Kampala (Central, Kawempe, Makindye, Nakawa & Rubaga). Also, other images were collected from Kyambogo University, Kampala. The study area was selected because it's the nature of the roads depicts the road network in the whole country. Also, the dataset was collected in different environmental conditions:

Table 1 Model evaluation metrics

Metric	Value
Mean average precision (mAP)	0.856
Precision	0.873
Recall	0.849

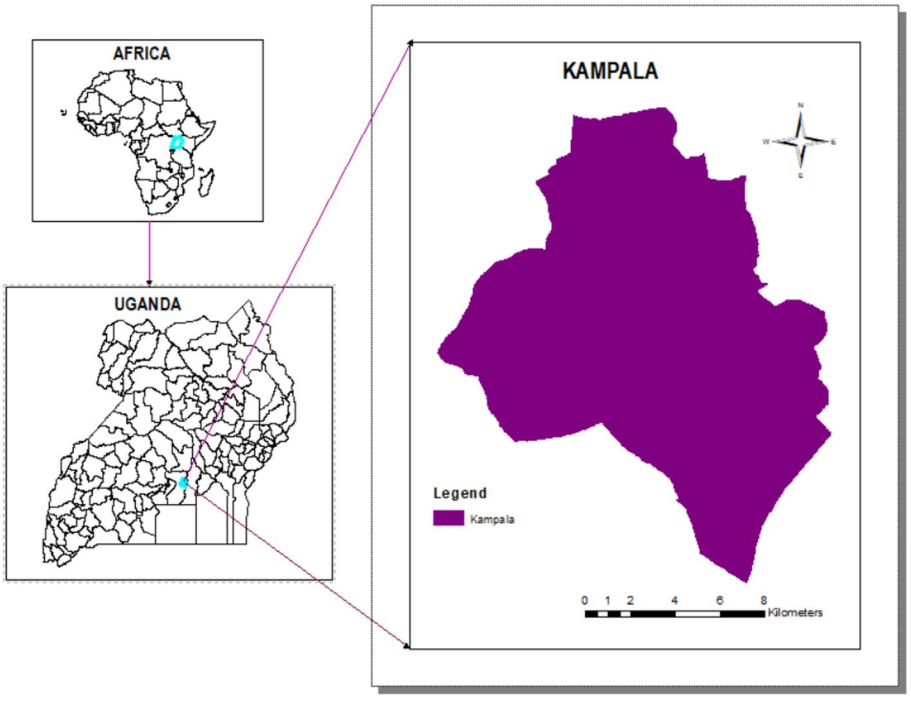

Fig. 3 Data Collection Area

variations in lighting, weather conditions (rainy, sunny, overcast), and time of day (morning, afternoon, evening). Besides, a subset of images was captured in indoor controlled environments to simulate real road scenarios where variations in lighting conditions and occlusions may occur, such as tunnels, underpasses, and shaded areas. Indoor data further makes the model robust by improving it under changing light conditions and most importantly when roads are poorly lit; this offers much better generalization in a practical application on many roads, part of which is badly illuminated-say, underground road parking or even covered highway routes or in a shadowy metropolis.

3.2 Dataset description

The dataset used for this study was designed to prioritize pothole detection while incorporating additional road-related objects to enhance contextual awareness and improve the model's robustness in real-world scenarios. The dataset comprises 11 object classes, including potholes, vehicles, signposts, walkways, and other relevant road features, as shown in Fig. 4. However, potholes remain the primary focus of detection, with other classes serving as complementary elements to reduce false positives and enhance model generalization. A total of 700 RGB images were collected from real-world road environments in Kampala, Uganda, covering diverse conditions such as lighting variations (morning, afternoon, evening, and low-light environments), different weather conditions (sunny, overcast, and rainy), and varied road textures (smooth asphalt, worn-out pavement, and gravel roads). Data collection was conducted using a smartphone camera (Infinix Hot 40i) with a 50MP dual AI rear camera to ensure high-resolution imagery. The camera was handheld at an approximate height of 1.2 to 1.5 m, simulating a pedestrian or vehicle viewpoint. To ensure that pothole detection remains the core objective, the dataset was structured with specific strategies. Class balancing was applied, ensuring

Fig. 4 Some of the Image datasets used during this study

that potholes had the largest representation in the dataset to achieve high precision in detection. Other road-related objects, such as vehicles and signposts, were included solely to enhance contextual differentiation rather than divert focus from pothole detection. The dataset was manually annotated using Roboflow in the YOLOv11 format. Bounding boxes were carefully placed around potholes to prevent confusion with road shadows, surface cracks, and irregular textures. Additional object annotations were used to improve the model's ability to distinguish potholes from other road features, reducing false positives. The dataset was then split into 80% for training, 10% for validation, and 10% for testing, ensuring a balanced evaluation of the model's performance.

3.3 Data preprocessing and labeling

To enhance the model's generalization ability and mitigate some of the problems caused by the dataset limitation, several preprocessing and data augmentation techniques were carried out. The original dataset was resized to 640×640 pixels to maintain compatibility with the detection model. This resolution was chosen for an appropriate trade-off between feature retention and computational efficiency. It is also the standard input size of YOLOv11. Auto-orientation was adopted to correct those misaligned images that were shot from different smartphone orientations during the process of collection. For increasing diversity in the dataset and improving model robustness, all of the following augmentation techniques were used to create three augmented versions per original image: Horizontal Flip (50%): Introduced to help the model recognize objects in mirrored orientations, which frequently occurs in real-world driving. Random Rotation (-15° to +15°): The augmentation applied simulates the variability of road inclinations and viewing angles to improve the model's invariance for object detection on minor orientations. Random 90° Rotations (Clockwise, Counterclockwise, Non-Equal Probability): To account for the possible perspective distortions that may arise because of camera angles, particularly while capturing potholes and other road textures. Brightness adjustment (between −15% to +15%): emulating changing lighting conditions such as bright daylight, overcast weather conditions, and interchanging shaded roadway sections, make

sure the model is invariant to illumination conditions. Gaussian blur: between 0 and 2.5 pixels: blur simulating due to motion as well as defects in camera focusing; helps a model generalize somewhat out-of-focus images. Salt and Pepper Noise (0.1% of pixels): For adding simulated sensor noise and low-light image artifacts that enhance the detection performance under bad weather conditions, such as fog or nighttime. The parameters used in augmentation were selected based on real variations in road conditions so that, under deployment, the model would generalize well in different environmental scenarios. These augmentation techniques specifically accounted for the usual challenges in outdoor pothole detection: lighting changes, angular variability, and motion distortion. Besides, annotation of images is done using Roboflow: https://universe.roboflow.com - one of the most popular annotation platforms in computer vision. Annotation has been done in the YOLOv11 annotation format, where every image includes a bounding box and class label. The final dataset was divided into training, validation, and testing sets, respectively, in a ratio of 80:10:10 for appropriate model evaluation. It is organized so that the configuration file pothole.yaml, along with the dataset of labeled data is easily integrated into the training pipeline of YOLOv11.

3.4 The proposed model

The proposed model unifies YOLOv11 with continuous learning to construct an adaptive object detection system for pothole detection in real-time. Unlike typical models, the proposed one doesn't need periodical retraining with newly arrived data, the proposed approach can dynamically learn novel road conditions incrementally using Class-Incremental Object Detection, while preserving all previously acquired knowledge. It performs this through experience replay: a revisit of previous samples, together with new data, will avoid catastrophic forgetting. Furthermore, it ensures that a proper balance in the model for new and historical knowledge is considered during training. The framework of continuous learning enables the model to adapt to changed environmental conditions through natural evolution independently of human involvement, such as lighting, textures of roads, and infrastructure. This makes it highly suitable in real-world settings within dynamic environments, including but not limited to road infrastructure monitoring and autonomous navigation, where such conditions are to be expected as evolving. The proposed model consists of a loss detection function, Continuous Learning Loss, Regularization Term (Memory Preservation), Total Loss Function, and Model Update Rule, as explained below.

3.4.1 Loss detection function

The YOLOv11 detection loss consists of three components:

$$L_{\text{det}}\left(X_t,\, Y_t;\, \theta\right) = L_{cls} + L_{loc} + L_{conf} \qquad (1)$$

Where:

$\boldsymbol{X_t}$ represents the input image or frame processed by the YOLOv10 model at time t.

$\boldsymbol{Y_t}$ is the corresponding ground truth label for the input image X_t.

θ: refers to the set of learnable parameters in the YOLOv10 model.

$\boldsymbol{L_{cls}}$ Classification loss (Cross-Entropy).

$\boldsymbol{L_{loc}}$ Localization loss (IoU or GIoU).

$\boldsymbol{L_{conf}}$ Confidence score loss (Binary Cross-Entropy).

3.4.2 Continuous learning loss

The continuous learning objective updates the model based on newly labeled data, weighted by the importance of recent samples:

$$L_{CL}(\theta) = \sum_{i=1}^{T} \alpha_i D(f(X_i; \theta), Y_i) \quad (2)$$

Where:

T: Current time index.

$\boldsymbol{\alpha_i}$ Sample weight, defined using an exponential decay function.

$$\alpha_i = e^{-\beta(T-i)}$$

Where β is a decay constant controlling how past data is weighted.

$D(f(X_i;\theta), Y_i)$ represents the detection loss function, which measures the discrepancy between the model's predicted output $f(X_i;\theta)$ and the true labels Y_i.

3.4.3 Regularization term (memory preservation)

To prevent forgetting previously learned data, Elastic Weight Consolidation (EWC) is applied:

$$L_{reg}(\theta) = \left(\frac{\lambda}{2}\right) \sum_i F_i (\theta_i - \theta_i^*)^2 \quad (3)$$

Where:

$\boldsymbol{F_i}$ Fisher Information Matrix (parameter importance).

$\boldsymbol{\theta_i^*}$ Previously learned optimal parameter values.

λ: Regularization strength.

3.4.4 Total loss function

The overall loss function becomes:

$$L(\theta) = L_{CL}(\theta) + L_{reg}(\theta) \quad (4)$$

Expanding this gives:

$$L(\theta) = \sum_{i=1}^{T} \alpha_i D(f(X_i; \theta), Y_i) + \left(\frac{\lambda}{2}\right) \sum_i F_i (\theta_i - \theta_i^*)^2 \quad (5)$$

3.4.5 Model update rule

The model parameters are updated using gradient descent:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta) \quad (6)$$

Where:

$$\nabla_\theta L(\theta) = \sum_{i=1}^{T} \alpha_i \nabla_\theta D(f(X_i; \theta), Y_i) + \lambda \sum_i F_i \left(\theta_i - \theta_i^*\right) \quad (7)$$

3.5 The architecture of the proposed model

The architecture of the proposed model in Fig. 5 integrates YOLOv11 with the concept of continual learning for object detection in dynamic environments. It contains three backbones: Backbone, Neck, and Head, combined within a continual learning framework to attain adaptability without catastrophic forgetting. Each of these components works together in harmony, capitalizing on the power of state-of-the-art feature extraction, fusion, and learning mechanisms that improve the accuracy and efficiency of detection. It contains the Backbone as a feature extraction module, which takes an input image resized to $640 \times 640 \times 3$ dimensions. A series of convolutional layers with growing channel depths is applied: 32, 64, 128, 256, and 512, capturing spatial and contextual information at multiple levels of abstraction. The network backbone is further integrated with enhanced CSP, or Cross-Stage Partial, blocks, denoted as C3k2, which improve feature fusion while reducing computational costs. The Spatial Pyramid Pooling-Fast module is adopted at the tail of the backbone to pool the features from different scales, enriching the model with multiscale contextual information. This hierarchical approach ensures that the backbone effectively captures features from small to large objects under various conditions. The Neck designs feature aggregation and fusion to enable the integration of multi-scale information for better detection. This backbone couples feature maps from different levels through concatenation layers to guarantee the contribution of both high-resolution and low-resolution features in the process of detection. Upsampling layers are included for aligning of the feature maps in lower resolution with the higher resolution ones by ensuring that fine-grained details are retained by the model. Besides, the neck is powered with advanced convolutional layers with parallel spatial attention, called C2PSA blocks. These blocks enhance the important spatial regions, thereby improving the model sensitivity to some critical features, such as object edges or fine details for tasks involving pothole detection. The Head outputs the final object predictions, including object classes, bounding box coordinates, and confidence scores. The detection layers in the head make use of the refined and fused feature maps to offer more precise identification and localization of objects present in an input image. This module ensures that the model can generate very accurate bounding boxes along with class probabilities, suitable for applications requiring high accuracy, such as real-time road monitoring. The most salient novelty of the proposed model is the Continual Learning Framework, which enables it to adapt dynamically to new, incoming data and an ever-changing environment. During training, the model learns incrementally from multiple tasks (1, 2,…,), with each task adding new data or object classes. A memory preservation strategy is

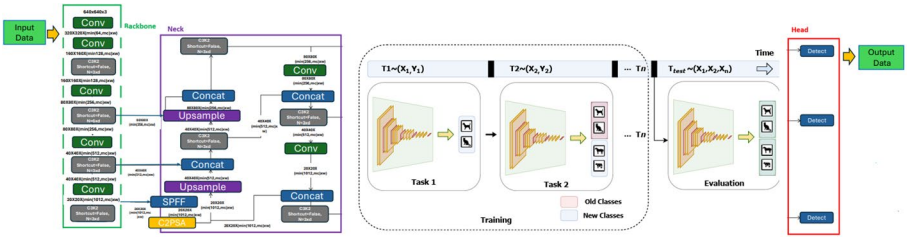

Fig. 5 The Architecture of the proposed model

followed, such as Elastic Weight Consolidation (EWC), which penalizes large changes in important model parameters to avoid catastrophic forgetting. This also ensures that the knowledge about previous tasks does not get lost while the model trains on new ones. At inference time, it scores test data based on knowledge accumulated over all the training tasks, hence enabling robust predictions in dynamic environments. A model with better performance could have been achieved for the proposed model by enhancing it with CSFF in the backbone and neck to integrate multi-scale features at different scales, further improving detection accuracy for objects of varying sizes. Besides, the architecture optimization for real-time performance also made it achieve high inference speeds suitable for the deployment of resource-constrained devices. Thus, it is effective for actual scenarios, such as Ugandan pothole-detecting tasks, where dynamic road conditions need adaptive and efficient object-detecting systems.

3.6 Model implementation

All of the experiments conducted for this study were performed using Google Colab. Additionally, a laptop running Windows 10 with an Intel Core i7 processor running at 2.50 GHz, 16GB of RAM, and a 2GB NVIDIA GeForce MX150 graphics card was used. Throughout the experiments, the default settings for YOLOv10n were maintained. Furthermore, 250 epochs were used in every experiment.

3.7 Performance metrics

The proposed model was evaluated using three performance metrics. These are Mean average precision (mAP), precision, and Recall, shown in Eqs. 8–10, respectively. Mean average precision (mAP): mAP is the average of AP scores over a set of object categories. It is a more comprehensive measure of accuracy than average precision (AP), as it takes into account the performance of the model on all object categories. Precision measures the accuracy of positive predictions. High precision means the model identifies the right objects most of the time, minimizing false positives. Recall measures the completeness of positive predictions. A high recall means the model misses very few real objects, minimizing false negatives.

$$mAP = \frac{1}{N} \sum {}^{N}_{i=1} AP_i \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

Where in mAP, N is the number of queries, and AP is the Average Precision for each class. In Eqs. 9 & 10, TP, TN, FP, and FN stand for true positive, false positive, and negative, respectively.

3.8 Model training

For this work, 250 epochs are utilized for the model training; such a quantity will help the network converge appropriately for YOLOv11 but not too quickly to avoid model convergence before getting meaningful representation. This was done based on practical

experimentation whereby lower numbers of epoch training revealed poorer detection accuracies, whereas additional training beyond that showed minor performance improvements while rapidly increasing computational cost. Various techniques were used to avoid overfitting, including early stopping, which monitored validation loss and stopped training if there was no improvement for 20 consecutive epochs. Also, L2 regularization-weight decay of 0.0005 was used on the model's parameters to prevent high complexity and thus improve generalization. We utilized an empirical initial learning rate of 0.001, tuned through trial experiments and grid search optimization, which helped in optimizing the model for better convergence and performance. To avoid oscillations and ensure stable convergence, we introduced a learning rate decay strategy where the learning rate is reduced by a factor of 0.1 every 50 epochs. This adaptive learning rate scheduling helps in avoiding overshooting the optimal solution while maintaining adequate learning momentum during later epochs. Apart from that, cosine annealing was added to create a gradual decrease in the learning rate, which would yield a fine-tuning of weight updates toward the end of training. Such optimization gave a faster convergence and better generalization, hence preventing overfitting. Dropout layers were also included in the network to reduce reliance on specific neurons so that when applied to real-world situations, the network would be better at handling them. For incorporating more variability into the training samples, augmentation techniques such as random rotations, flips, changes in brightness, and noise perturbation were used in order to generalize the model to different road conditions, lighting, and weather.

3.9 Model validation

Comprehensive model validation was necessary to make the proposed continuous learning YOLOv11 model robust and reliable. In this respect, the dataset was divided into three parts: 80% for training, 10% for validation, and 10% for testing. This was done so that there was enough data to train the model, but with an adequate number of validation and test sets to assess the generalization performance of the model. The training has been carried out by monitoring and fine-tuning the validation set, while the test set is used for testing the performance of the model on new data. These models were evaluated using the common metrics that include mean average precision, precision, and recall. The mAP provides a comprehensive score of the model's accuracy by taking an average of the precisions of the object classes. Precision, as the ratio of true positives to all positive predictions, may be seen as the model's ability to reduce false positives. Recall, on the other hand, presents the true positives share in actual positive cases across the dataset. It shows a model's capability to avoid false negatives. These metrics showed clear, broad measurements of performance for the models based on the detection task. It was utilized during the training process for observing performance and tuning hyperparameters of the model. Early stopping was used as a technique to prevent overfitting by automatically stopping the training process after the validation loss stopped improving. Besides that, learning rate scheduling dynamically adjusts the learning rate concerning validation performance and guarantees optimal convergence and much faster training. These techniques helped in enhancing the stability and efficiency of the training process. The final performance of the model was evaluated on the test set, which consisted of unseen data. Quantitative metrics such as mAP, precision, and recall were calculated to assess the model's accuracy in detecting objects under varied conditions. Furthermore,

class-wise evaluations were done to understand the model's performance on specific object classes, such as potholes, vehicles, and signposts. The presented analysis provided the model's appropriateness to different real road monitoring conditions and its capability to adapt under changing conditions. Hence, the model is reliable, robust, and ready for deployment after following such an elaborate validation process in dynamic environmental conditions.

4 Experimental results

4.1 Performance on mAP, Precision, and recall

The performance of the proposed model was evaluated in accordance with Mean Average Precision at 50% IoU, Precision, and Recall-standard metrics of object detection that measure the accuracy, reliability, and detection efficiency of the algorithm. mAP@50 refers to the mean of the average precision scores for all detected objects at an IoU threshold of 50%. This metric will evaluate how well the model is correctly localizing and classifying objects while ensuring minimal false detections. A higher mAP@50 indicates that the model has been detecting objects and positioning them within bounding boxes that highly overlap with the actual objects. The mAP@50 score of 0.856 suggests that the model provides reliable detections, successfully identifying potholes with high accuracy. Precision is 0.873, and it reflects the ratio of correctly identified potholes with respect to the overall number detected. This ensures that the generation of false positives by the model is really low. Recall gives the proportion of actual potholes correctly found, while 0.849 says it can find the majority of them, hence reducing the false negatives. The close fit between Precision and Recall values indicates a balanced model that neither misses critical detections nor triggers excessive false alarms. This qualifies the model for real-time road monitoring, where the accuracy of detection and completeness both go hand-in-hand to realize effective pothole detection and management of road infrastructure.

4.2 Visualization of the performance with epochs

Figure 6 shows the evaluation of Precision, Recall, and mAP@50 over 100 epochs of training, reflecting how the model learns over time. At the beginning, all three metrics start at a relatively lower value: Precision ~ 0.5, Recall ~ 0.45, and mAP@50 ~ 0.48. As the training progresses, the metrics keep improving with a steady trend in the learning and optimization of the model. After around epoch 70, all three metrics stabilize and approach their peak values, indicating convergence. Specifically, Precision reaches approximately 0.873, Recall stabilizes at 0.849, and mAP@50 reaches 0.856 at the 90th epoch, where the performance plateaus. The slight flattening of the curves towards the final epochs suggests that further training provides diminishing returns since the model has likely reached its optimal performance. The close alignment of precision and recall values indicates a balanced trade-off where the model performs equally well in minimizing false positives and maximizing true positive detections. The convergence and balance here point toward the effectiveness of the model in learning robust features for accurate pothole detection.

4.3 Detection of different objects in a road with potholes

The class-wise performance of the model elaborates on the strength of the model in the detection of different objects in the dataset. Table 2: Precision, Recall, mAP@50 for

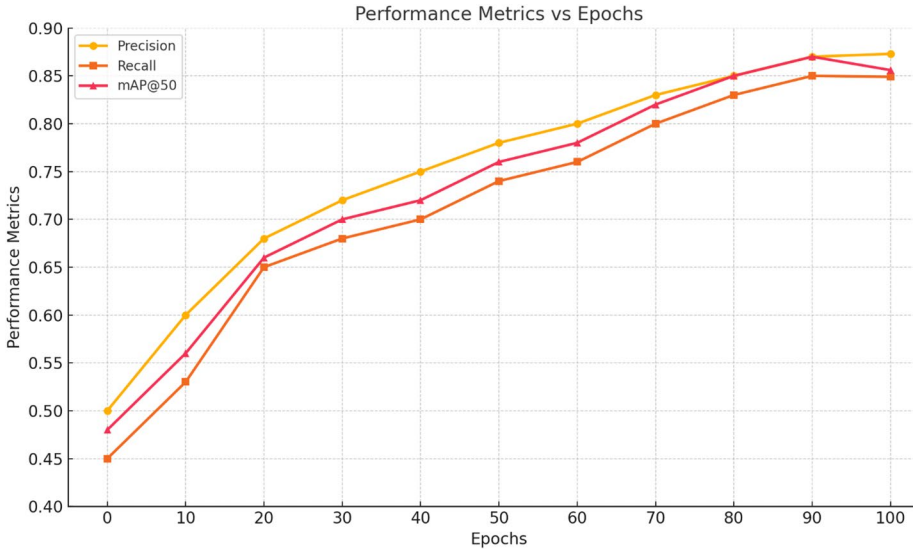

Fig. 6 Evaluation of Precision, Recall, and mAP@50 over 100 epochs

Table 2 Class-Wise evaluation results

Class	Precision	Recall	mAP@50
Stairs	0.72	0.75	0.78
Walk way	0.81	0.83	0.84
Trees	0.65	0.67	0.7
Bicycle	0.78	0.81	0.82
Boda	0.79	0.82	0.83
Building	0.82	0.85	0.86
Cars	0.85	0.88	0.89
Fence	0.76	0.78	0.8
Pole	0.74	0.77	0.79
Sign post	0.71	0.74	0.76
Pothole	0.84	0.86	0.88

various classes: potholes, cars, buildings, trees, stairs, fences, and walkways. It performs exceedingly well on the detection of potholes-0.88 mAP@50, 0.89 mAP@50 for cars, and 0.86 mAP@50 for buildings-suggesting its reliability in finding these key objects with minimal false alarms or misclassifications. The model, therefore, learns the distinctive characteristics of well-structured and frequently occurring objects within the dataset. However, it showed a little lower mAP@50 for trees at 0.70 and stairs at 0.78. Tree detection is problematic due to variations in appearance, occlusions, and complex blending with the background. In most urban scenarios, trees have non-rigid, diverse structures and may be partially occluded by other objects like vehicles and buildings lining either side of the street. Moreover, changes in lighting conditions and seasonal changes in foliage density may result in variations in the appearance of trees, which will make it difficult for the model to consistently detect them. Similarly, stairs detection proved less reliable; this was probably due to differences in perspective, occlusions, and visual similarity to other road infrastructure elements, such as sidewalks and pedestrian pathways. It is possible that the model fails to distinguish stairs from walkways, especially in poor lighting conditions or when viewed from oblique camera angles. There are also fewer examples of stairs in the dataset, which could lead to insufficient training examples,

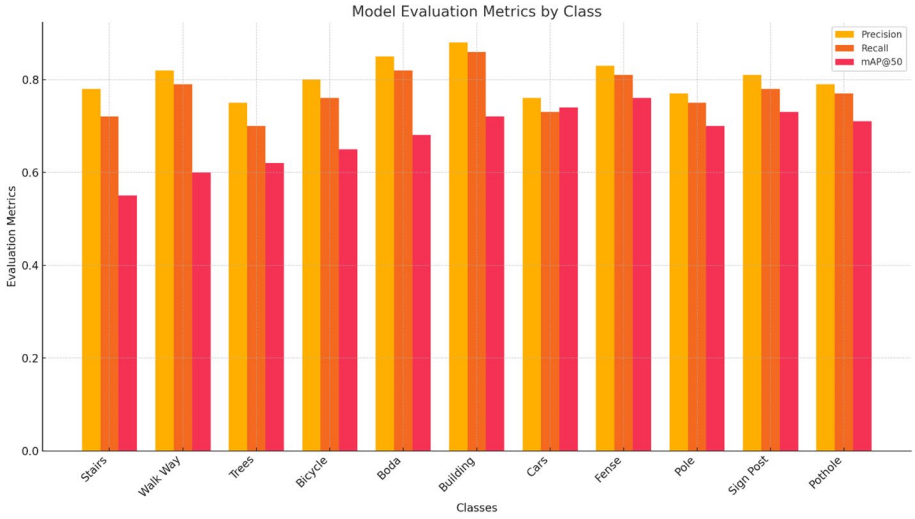

Fig. 7 Precision, Recall, and mAP@50 for different object detections

Table 3 Inference speed

Metric	Value
Average inference time (ms)	12.7
Frames per second (FPS)	78.7
Batch size used	16.0

reducing the generalization capability of the model for this class. The low performance of these particular classes points out ways for future improvements. Further improvements can be made by gathering more diverse data on classes of trees and stairs, improving augmentation techniques for specific classes, or even using mechanisms of adaptive learning of features that refine the representation of objects. Despite this, the model has remained highly effective in detecting potholes and other critical elements of roads; hence, quite suitable for application in real-time road monitoring. In the future, efforts will be geared toward improving these limitations for better detection performance on all object categories.

4.4 Visualization of the model performance per class

Figure 7 presents Precision, Recall, and mAP@50 for different object detections, such as the critical classes of potholes, cars, and buildings. The model exhibits good performance in car, building, and pothole detection, with Precisions, Recalls, and mAP@50 all being greater than 0.8. The big difference in values for tree and stair classes, going from low to about 0.6 in mAP@50, indicates poor detections of these classes. Thus, this may indicate issues with the accurate detection of these objects due to partially overlapping features, complex scenes in general, or variability in how they appear. While similar, the classes of fence and sign post show moderate detection accuracy, with metrics averaging at about 0.75, reflecting some inconsistencies likely because of smaller object sizes or occlusions.

4.5 Inference speed of the proposed model

The results in Table 3 show that the proposed model achieves an average inference time of 12.7 milliseconds per image, which is equivalent to a performance of 78.7 frames per

second. It is within a very comfortable range for real-time applications, given that studies [54] have shown that at least 30 FPS is usually acceptable for most real-time object detection tasks, especially in the domain of autonomous driving or when operating surveillance systems. The achieved FPS is indicative of the model's capability to process a large volume of frames within a very short time, hence highly applicable for time-sensitive applications like road monitoring. Also, the batch size of 16 used for the inference underlines the efficiency of the model in handling multiple inputs without significant latency.

4.6 The hyperparameters of the proposed model

The hyperparameters selected for training the proposed model are carefully chosen to obtain an optimal trade-off between accuracy and computational efficiency as shown in Table 4. A learning rate of 0.001 was chosen for smooth and optimal convergence, which avoids overshooting or extremely slow training. For the batch size, 16 was chosen, which is an optimum balance between computational efficiency and memory usage and enables smoother gradient updates. The model was trained for 100 epochs to give it ample time to learn while simultaneously adopting precautions against overfitting. The input image size was kept at 640×640 pixels to capture the minute details necessary for detection without being computationally burdensome. The Adam optimizer was used since it adapts the learning rate by utilizing momentum and RMSprop techniques, hence improving efficiency and stability in training. First, momentum with a value of 0.9 was used to push the updates in the correct direction to speed up convergence. Then, weight decay was set to 0.0005, which would penalize large weights, hence avoiding overfitting and resulting in better generalization. Other considerations included a regularization strength of 1.0 and a memory buffer size of 5000 samples, ensuring that the storage and retrieval of training data were efficiently handled. The weights of the model were updated every 10 epochs, further stabilizing the training process. These hyperparameter choices collectively contributed to the model's robust performance across various evaluation metrics, hence proving to be suitable for real-world pothole detection in Table 4.

4.7 Results on object detections

The results in Fig. 8 show results on the detected objects. The proposed ensemble model was fine-tuned for object detection in the local environment in Uganda. This is because we plan to deploy our model to our local setting to help people with blind impairments in their movements, the collected images were from an environment where blind people

Table 4 Hyperparameters

Hyperparameter	Value
Learning rate (η)	0.001
Batch size	16
Epochs	100
Image size	640×640
Optimizer	Adam
Momentum	0.9
Weight decay	0.0005
Sample weight decay (β)	0.05
Regularization strength (λ)	1.0
Memory buffer size	5000 samples
Update frequency	Every 10 epochs

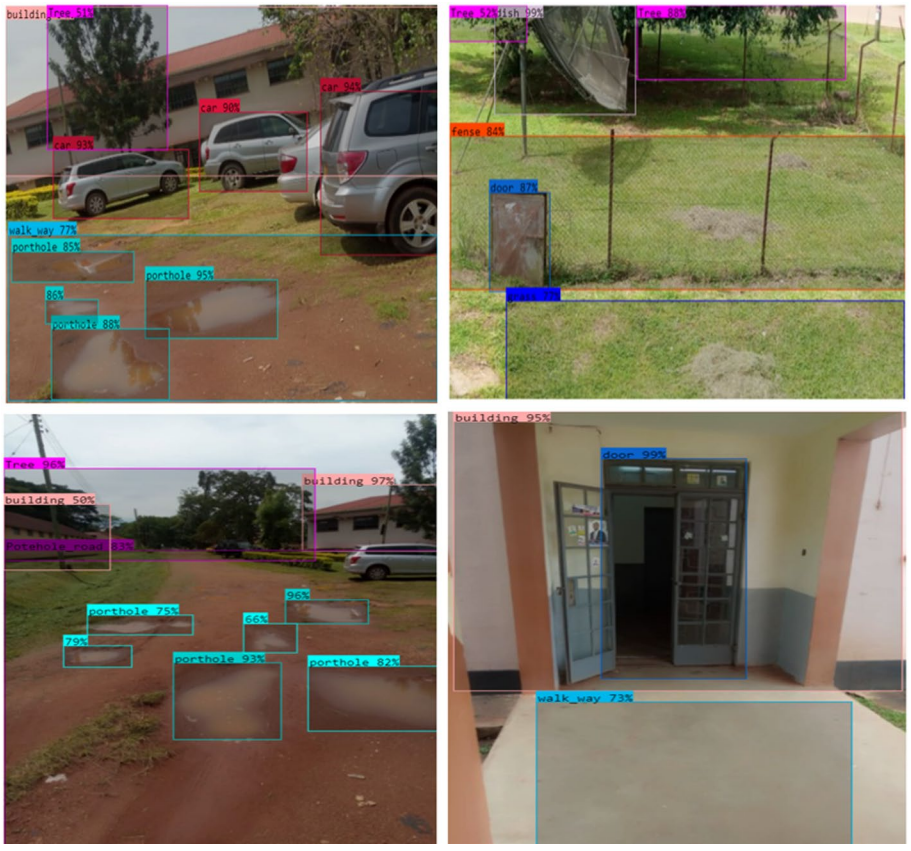

Fig. 8 Results on Object Detections

Table 5 Ablation study results for different training configurations

Training configuration	mAP@50 (Potholes)	Precision	Recall
Single-class pothole detection	0.832	0.841	0.827
Dual-class pothole detection	0.868	0.874	0.856
Multi-Class road obstacle detection	0.849	0.865	0.842

move and interact with their peers and families around Kyambogo University. The model effectively detects a diverse set of obstacles, ensuring safer navigation in real-world environments. As shown in the Fig. 8 below, the detection of stairs, trees, and walkways is essential for assistive mobility applications, demonstrating the adaptability of the proposed model beyond pothole detection. The results reveal that our proposed model was able to detect bad roads (pothole roads) that are common in Uganda. The model also detected objects such as walkways, buildings, trees, stones, and other objects, as shown below.

4.8 Ablation study: impact of Multi-Class training on pothole detection

To validate the effect of including multiple classes in the dataset on pothole detection performance, an ablation study was conducted, see Table 5. This analysis aimed to assess whether training the model with additional classes improves or diminishes its effectiveness in detecting potholes. The study compared the model's performance under three different training configurations. The first configuration involved Single-Class Pothole Detection, where the model was trained exclusively on the "pothole" class, treating all

road defects as a single category. The second configuration, Dual-Class Pothole Detection, used two distinct pothole-related classes: "pothole" (individual potholes) and "pothole_road" (severely degraded road surfaces with multiple potholes). The third configuration, Multi-Class Road Obstacle Detection, trained the model on all object classes, including potholes, vehicles, trees, walkways, and other road-related features. To ensure a fair comparison, the dataset size remained consistent across all configurations. Each model was trained for 100 epochs using the same hyperparameters, including a batch size of 16, a learning rate of 0.001, the Adam optimizer, and an input image size of 640×640 pixels. Performance was evaluated using mean average precision (mAP@50), precision, and recall as key metrics.

The results indicate that using a single pothole class resulted in the lowest mAP (0.832), precision (0.841), and recall (0.827). This suggests that merging all pothole-related damage into a single category limited the model's ability to differentiate between varying degrees of road degradation. In contrast, introducing the "pothole_road" class improved mAP to 0.868, precision to 0.874, and recall to 0.856. This demonstrates that distinguishing between individual potholes and widespread road damage helped the model make more accurate predictions, reducing false positives and false negatives. When trained on all object classes, the model's pothole detection mAP slightly dropped to 0.849, while precision and recall remained competitive at 0.865 and 0.842, respectively. This indicates that while adding more classes improves general obstacle detection, it slightly affects pothole-specific accuracy. The increase in accuracy when using two pothole-related classes can be attributed to better class differentiation. In real-world scenarios, roads may have isolated potholes or large, degraded sections with numerous potholes. When both cases are treated as a single class, the model struggles to generalize effectively, leading to misclassification. By explicitly distinguishing between "pothole" and "pothole_road," the model can learn more refined features for different types of road damage. On the other hand, training the model on multiple obstacle classes slightly reduced pothole detection accuracy. This is likely due to increased class competition in feature learning. However, the trade-off is that the model becomes more adaptable for broader road monitoring applications beyond just pothole detection. The findings from this ablation study suggest that for highly specialized pothole detection applications, using only the "pothole" and "pothole_road" classes is optimal, as it yields the highest accuracy. However, for general road infrastructure monitoring, including additional object classes (e.g., vehicles, walkways, trees) provides better versatility while maintaining competitive pothole detection performance.

4.9 Practical implications of model performance in real-world scenarios

The proposed model achieves high performance metrics, which also include an mAP of 0.856, precision of 0.873, and recall of 0.849, showing a strong capability in detection. In real-world applications, especially on roads, timely and proper pothole detection plays an important role in reducing road accidents and managing the infrastructure effectively. A high precision score of 0.873 in this regard would mean minimum false positives, that is, pothole detection where it does not exist. This aspect is very crucial for the onboard warning system in vehicles and also for road maintenance planning, since false alerts will degrade the trust of users in such systems. Second, a recall score of 0.849 says a lot about how effective the model is in identifying true potholes, hence preventing

Table 6 Performance comparison of the proposed model with State-of-the-Art approaches

Model	mAP@50	Precision	Recall	FPS	Key Features	Limitations
YOLOv10 [5]	0.823	0.841	0.821	65.4	Strong baseline accuracy, efficient on common datasets	Lacks adaptive learning, retraining required
LD-YOLOv10 [6]	0.811	0.833	0.806	73.5	Optimized for edge devices, compact architecture	Limited adaptability to dynamic classes or new domains
Tiny-DSOD [8]	0.795	0.812	0.798	70.2	Lightweight, good for low-power deployments	Weaker performance for small/occluded object detection
Proposed YOLOv11 + CL	0.856	0.873	0.849	78.7	Adaptive learning, strong detection under dynamic scenes	Higher initial computational demand than Tiny-DSOD

hazardous road conditions. This reduces the risk of accidents involving undetected potholes, especially for motorcyclists and cyclists, who are more susceptible to road surface irregularities. Real-time detection capability will, therefore, enable the model to proactively monitor systems that would allow local authorities to prioritize repair efforts based on the severity and distribution of potholes. Integration of the detection system with edge devices and IoT-based infrastructure may support automated reporting systems that notify maintenance teams the very moment new potholes appear. This can result in cost-efficient and timely road repairs, reducing long-term degradation of infrastructure and optimizing resource allocation for road maintenance projects. The resultant FPS of 78.7 proves that the model can be feasible for real-world deployment in self-driving cars, smart city monitoring, and moving inspection vehicles. Given the deficiency in the identification of some classes, such as trees and stairs, further optimization will increase the adaptability to various road settings. Despite these, the model presents a scalable, efficient solution to improve road safety and maintenance operations, hence demonstrating a high potential for large-scale implementations in developing countries with limited roads and monitoring infrastructure.

4.10 10 comparative evaluation and model robustness

To ensure the reliability, adaptability, and competitiveness of the proposed YOLOv11-based model with continuous learning, a comparative analysis was conducted against three leading state-of-the-art object detection approaches: YOLOv10 [5], LD-YOLOv10 [6], and Tiny-DSOD [8], as shown in Table 6. These models have been extensively used in road infrastructure monitoring, vehicle detection, and real-time edge deployment, making them ideal baselines for performance benchmarking. The proposed model integrates continuous learning mechanisms to adapt to new obstacles in dynamic road environments, reducing the need for frequent retraining. It also incorporates multi-scale feature fusion and attention-enhanced modules, improving detection of small, occluded, and irregularly shaped objects like potholes. The comparison was based on four key metrics: mean average precision (mAP@50), precision, recall, and inference speed (frames per second - FPS), evaluated under consistent experimental conditions. The results in Table 6 highlight the superior performance of the proposed model across all metrics. With a mAP@50 of 0.856, precision of 0.873, and recall of 0.849, the model demonstrates high accuracy in both object localization and classification. Furthermore,

it achieves an inference speed of 78.7 FPS, validating its real-time processing capability, even in resource-constrained environments.

5 Discussions

The proposed model achieved high detection performance: mAP of 0.856, precision of 0.873, and recall of 0.849. These results further strengthen the observation of recent developments in the real-time object detection arena, showing that the optimized variants of YOLO have significantly improved detection accuracy along with computational efficiency. Various works such as [5, 6], and [7] have identified the requirement for lightweight architecture toward real-time applications, especially when the resources are at a premium. In comparison, the current model, optimized for edge-device deployment in LD-YOLOv10 [8], is now able to allow for continued learning by letting the model evolve with road conditions dynamically without needing frequent retraining. The proposed model, therefore, contributes to road infrastructure monitoring by proposing a scalable, adaptive, real-time pothole detection framework.

Pothole detection is also very important from the viewpoint of road safety, as it ensures minimum vehicle damage and reduces accident risks. Various works [4] and [9] have highlighted the importance of a strong object detection system for autonomous driving and traffic management. The high precision, 0.873, assured in the present model, guarantees close to zero numbers of false alarms and, accordingly, any unjustified intervention by autonomous systems. Recall here measures 0.849, representing the degree to which actual instances of road surface hazards have been identified correctly for proactive maintenance, thus establishing sufficient grounds to deploy this model in a smart transport system; AI-driven data could enable the road authorities to plan for selective road repairs based on the same.

Performance in real-world object detection is one of the most challenging tasks and usually involves many complicated situations like occlusion, changes in lighting conditions, and weather dynamics. In this respect, multiscale feature fusion, spatial attention, and domain adaptation methods have been shown to perform well in improving detection robustness by existing literature [44, 50]. It performs well on the different classes of road objects that are considered here and shows the robustness of obstacle detection: road, vehicles, and pedestrians. It can be very useful in urban monitoring systems where there is a huge infrastructure degradation, and continuous assessments are required.

One of the most important strengths of the model is leveraging continuous learning for real-world applications. The continuous learning framework does not require periodic retraining by humans, as traditional models do; it can automatically update the knowledge of the model itself with new, emerging road conditions. This is very valuable for smart road monitoring systems over a long period because the detection accuracy might largely degrade due to factors such as evolving seasonal weather effects, traffic, and repairs. Moreover, the robustness of the model means that the performance is stable over a long period and requires fewer human interventions. This further follows modern trends in edge AI and smart city initiatives, discussed in [10, 49], and [51].

5.1 Broader implications for road infrastructure monitoring

The adoption of AI-based road monitoring systems is one of the major steps toward sustainable infrastructure management. Poor road conditions lead to increased

maintenance costs, vehicular damage, and higher fuel consumption due to inefficient driving patterns [52]. Real-time pothole detection through AI-powered systems enables authorities to prioritize road repairs, reducing maintenance costs in the long run and improving transportation safety. Besides, considering embedding such models in UAV-based monitoring systems or smart city infrastructure, large-scale assessments of road conditions can be performed, which is discussed in [31]. Through the concept of continual learning, the model self-updates in detecting newer kinds of road anomalies and makes the model quite practical for long-term use. Another contribution of this research is creating assistive navigation for visually impaired people. However, by extending this model to identify obstacles other than potholes, such as stairs, sidewalks, and street barriers, the model can form the basis for AI-powered mobility aids. Several research works related to assistive technologies, including [53], indicate that real-time obstacle detection is a necessity to enhance mobility solutions for visually impaired pedestrians. Therefore, in future work, it will be interesting to test the feasibility of deploying the proposed model in smart wearable or handheld navigation devices.

5.2 Computational resource requirements for model deployment

The proposed road pothole detection model involves computationally intensive processes that need considerable processing power, memory, storage, network bandwidth, and energy efficiency. Considering the processing time of the frame, at 12.7 ms with a speed of 78.7 FPS, the proposed system could be appropriate to deploy only on high-end GPU architectures, which include recent architectures like NVIDIA RTX series 5000 onward, or the Jetson Xavier NX at the edge side. The computational complexity, driven by a 640×640 input resolution and multiple convolutional layers, requires hardware accelerators with parallel processing capabilities, such as Tensor Cores or AI-optimized ASICs, to achieve real-time performance. Memory requirements include 8–16 GB of VRAM for efficient GPU execution, with an additional 2–4 GB of system memory for handling continual learning processes, ensuring robustness and adaptability in dynamic environments. Storage requirements are moderate, about 10–20 GB SSD, including model checkpoints, logs, and datasets. Real-time applications will require high-bandwidth, low-latency 4G/5G connectivity to keep up with the high frame rates of continuous video streaming. Future deployments can also be made using next-generation GPU architectures like the RTX 5000 series or dedicated AI accelerators.

5.3 Societal and economic impact in developing regions

Deployment of advanced pothole detection in developing regions is of great societal and economic potential. In most developing countries, bad road conditions remain a big concern because they increase vehicle maintenance costs, raise accident rates, and impede effective transportation. With real-time pothole detection, timely maintenance will be possible to prevent these issues and ensure safer roads [55]. Improved road conditions contribute a lot to economic benefits. A well-connected transportation network is the backbone of trade and commerce; better-maintained roads imply less expensive operating costs for vehicles and travel time. Such efficiency could increase economic activities with the smooth movement of goods and people. Besides, predictive maintenance through real-time detection will provide optimal resource allocation to make sure that the limited fund is used effectively in road repairs [56]. From a social perspective,

better road safety from timely detection of potholes reduces accident rates and, hence, injuries and fatalities. It is a contribution to public health and safety, thus fostering more community well-being. Further, good road conditions may improve access to basic services such as healthcare and education, particularly in rural areas, thus contributing to social equity and development [57]. While the benefits are evident, there are also several challenges to deploying these systems in developing regions. All demands in the form of poor infrastructure, budget constraints, and capacity building need to be effectively addressed [58]. Cooperative approaches by government agencies, local communities, and international partners will bring about effective implementation. Overall, the implementation of real-time pothole detection systems in developing regions has huge potential for social and economic benefits. By maintaining better roads, these systems would improve safety on the roads, thus contributing to general development objectives such as a higher quality of life and economic well-being in those areas.

6 Conclusion

The proposed paper presents the optimized YOLOv11-based object detection model for real-time pothole detection with continuous learning while driving in Uganda. It performed very well during the experiments in detecting road potholes by achieving a high mean average precision of 0.856, precision of 0.873, and recall of 0.849. Class-wise performance evaluations indicate that the model has performed well for key objects related to road monitoring, with potholes achieving a mAP of 0.88, vehicles at 0.89, and buildings at 0.86. These are indications that indeed it can detect both static and dynamic obstacles to improve road safety and reduce accident risks caused by poor infrastructure. Unlike the traditional object detectors based on YOLO, the proposed model integrates continuous learning that can adapt to the evolution of road conditions without frequent retraining. This is an important feature in certain applications, such as autonomous road monitoring and smart transportation systems, where environmental changes are dynamic. It also shows that this model, in real-time performance, reaches 78.7 FPS, corresponding to an inference time of 12.7 milliseconds, hence confirming its feasibility for deployment on edge devices like the NVIDIA Jetson Xavier NX and making it suitable for resource-constrained environments. However, experimental validation on edge devices was beyond the scope of this work and will be explored in future research. This work goes beyond mere pothole detection and extends the contribution of the model in aiding visually impaired persons with real-time obstacle recognition and path guidance. The detection of a wide range of objects like roadblocks, sidewalks, and moving obstacles draws on its application relevance in smart mobility. Future work will focus on integrating this model into IoT-based assistive navigation systems to improve accessibility for visually impaired pedestrians. Comparisons with the state-of-the-art object detection models such as YOLOv10, LD-YOLOv10, and Tiny-DSOD demonstrate the competitive performance of the proposed model, especially on the detection of small and occluded objects. Additional focus on multi-scale feature fusion and adaptive learning mechanisms empowers it to be highly robust across diverse environmental concerns such as low illumination and occluded object detections. However, further benchmarking against edge AI models is required to prove its superiority for low-resource deployments. These results will, in general, provide great value for real-time monitoring of road infrastructure, AI-driven transportation systems, and assistive technologies for visually

impaired people. With its high accuracy, adaptability, and real-time processing capabilities, this model is very well positioned as a scalable solution for automated pothole detection, smart city monitoring, and AI-driven accessibility applications.

6.1 Limitations and future research directions

While the model achieves high accuracy and real-time detection, its deployment on edge AI devices remains untested. Future research will focus on optimizing it for low-power hardware like Jetson Nano. The dataset, primarily from Kampala, Uganda, limits generalization; expanding it to diverse environments is necessary. The reliance on RGB images restricts detection in low-light and adverse conditions; integrating LiDAR and thermal imaging can improve robustness. Additionally, testing its application for assisting visually impaired individuals is needed. Future work will explore sensor fusion, federated learning, and AI-powered navigation systems for broader real-world applications.

Author contributions
Nkalubo Lenard Byenkya (NLB): Writing the manuscript, Data Interpretation and Analysis, Methodology, Implementation, Manuscript Writing, and Review, read and approved the final version of the manuscript for publication. Nakibuule Rose (NR): Manuscript Review, supervision, and approval of the final version of the manuscript for publication. Okila Nixson (ON): Manuscript Review, and approval of the final version of the manuscript for publication. All authors have read and approved the final version of the manuscript.

Funding
We acknowledge the financial support from Kyambogo University.

Data availability
The data utilized in this study is available from the corresponding author up on request.

Code Availability
Code is available upon request from the corresponding author.

Declarations

Competing interests
The authors declare no competing interests.

Human Ethics and Consent to Participate
Not applicable.

Received: 16 December 2024 / Accepted: 31 December 2025
Published online: 13 January 2026

References

1. Zia H, Hassan I, Khurram M, Harris N, Shah F, Imran N. Advancing road safety: A comprehensive evaluation of object detection models for commercial driver monitoring systems. Futur Transp. 2025;5:1–18.
2. Zanule PG. Road management system and road safety in Uganda. Walden Univ, 2015.
3. Liang S et al. Object Detectors in the Open Environment: Challenges, Solutions, and Outlook, 2024, [Online]. Available: http://arxiv.org/abs/2403.16271
4. Nicolas H, Nicolas H. Object detection detection in low light environments. Procedia Comput Sci. 2024;246:3381–9. https://doi.org/10.1016/j.procs.2024.09.219.
5. Li Y, Li J, Lin W, Li J. Tiny-DSOD: Lightweight object detection for resource-restricted usages, *Br. Mach. Vis. Conf.* 2018, BMVC 2018, 2019.
6. Sundaresan Geetha A, Alif MAR, Hussain M, Allen P. Comparative analysis of YOLOv8 and YOLOv10 in vehicle detection: performance metrics and model efficacy. Vehicles. 2024;6(3):1364–82. https://doi.org/10.3390/vehicles6030065.
7. Geetha AS, Hussain M. A Comparative Analysis of YOLOv5, YOLOv8, and YOLOv10 in Kitchen Safety, pp. 1–17, 2024, [Online]. Available: http://arxiv.org/abs/2407.20872
8. Qiu X, Chen Y, Cai W, Niu M, Li J. LD-YOLOv10: A lightweight target detection algorithm for drone scenarios based on YOLOv10. Electron. 2024;13(16). https://doi.org/10.3390/electronics13163269.
9. Mercaldo F, Martinelli F, Santone A. Real-Time road sign localisation through object detection. Procedia Comput Sci. 2024;246:30–7. https://doi.org/10.1016/j.procs.2024.09.225.
10. Abba S, Bizi AM, Lee JA, Bakouri S, Crespo ML. Real-time object detection, tracking, and monitoring framework for security surveillance systems. Heliyon. 2024;10(15):e34922. https://doi.org/10.1016/j.heliyon.2024.e34922.
11. Hussain M, Khanam R. In-Depth review of YOLOv1 to YOLOv10 variants for enhanced photovoltaic defect detection. Solar. 2024;4(3):351–86. https://doi.org/10.3390/solar4030016.

12. Safyari Y, Mahdianpari M, Shiri H A review of vision-Based pothole detection methods using computer vision and machine learning. Sensors. 2024;24(17). https://doi.org/10.3390/s24175652.
13. Ali ML, Zhang Z. The YOLO framework: A comprehensive review of Evolution, Applications, and benchmarks in object detection. Computers, 2024.
14. Trigka M, Dritsas E. A comprehensive survey of machine learning techniques and models for object detection. Sens (Switzerland). 2025. https://doi.org/10.3390/s25010214.
15. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, https://doi.org/10.1109/CVPR.2016.91
16. Wang Z, et al. PC-YOLO11s: A lightweight and effective feature extraction method for small target image detection. Sensors. 2025;25(2). https://doi.org/10.3390/s25020348.
17. Sapkota R et al. YOLOv10 to Its Genesis: A Decadal and Comprehensive Review of The You Only Look Once (YOLO) Series, 2024, [Online]. Available: http://arxiv.org/abs/2406.19407
18. Menezes AG, de Moura G, Alves C, de Carvalho ACPLF. Continual object detection: A review of definitions, strategies, and challenges. Neural Netw. 2023;161:476–93. https://doi.org/10.1016/j.neunet.2023.01.041.
19. Shaheen K, Hanif MA, Hasan O, Shafique M. Continual learning for Real-World autonomous systems: Algorithms, challenges and frameworks. J Intell Robot Syst Theory Appl. 2022;105(1). https://doi.org/10.1007/s10846-022-01603-6.
20. Shi H et al. Continual Learning of Large Language Models: A Comprehensive Survey, *ACM Comput. Surv.*, vol. 1, no. 1, pp. 1–47, 2024, [Online]. Available: http://arxiv.org/abs/2404.16789
21. Guo B, Zhang H, Wang H, Li X, Jin L. Adaptive occlusion object detection algorithm based on OL-IoU. Sci Rep. 2024;14(1):27644. https://doi.org/10.1038/s41598-024-78959-2.
22. Liu W, Liu J, Su X, Nie H, Luo B. Source-free Domain Adaptive Object Detection in Remote Sensing Images, vol. 14, no. 8, pp. 1–14, 2024, [Online]. Available: http://arxiv.org/abs/2401.17916
23. Kotar K, Mottaghi R. Interactron: Embodied Adaptive Object Detection, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2022-June, pp. 14840–14849, 2022, https://doi.org/10.1109/CVPR52688.2022.01444
24. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-End object detection with Transformers. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 2020;12346:213–29. https://doi.org/10.1007/978-3-030-58452-8_13. LNCS.
25. Jeon M, Seo J, Min J. DA-RAW: Domain Adaptive Object Detection for Real-World Adverse Weather Conditions, *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 2013–2020, 2024, https://doi.org/10.1109/ICRA57147.2024.10611219
26. Khodabandeh M, Vahdat A, Ranjbar M, MacReady W. A robust learning approach to domain adaptive object detection. Proc IEEE Int Conf Comput Vis. 2019;2019–Octob:480–90. https://doi.org/10.1109/ICCV.2019.00057.
27. Su S, Chen R, Fang X, Zhang T. A novel Transformer-Based adaptive object detection method. Electron. 2023;12(3). https://doi.org/10.3390/electronics12030478.
28. Li W, Li F, Luo Y, Wang P, Sun J. Deep domain adaptive object detection: a survey. IEEE. 2020. https://doi.org/10.1109/SSCI47803.2020.9308604.
29. Liu C, Zhang S, Hu M, Song Q. Object detection in remote sensing images based on adaptive Multi-Scale feature fusion method. Remote Sens. 2024;16(5). https://doi.org/10.3390/rs16050907.
30. Saleh S, Jolfaei A, Tariq M. Real-Time pothole detection with edge intelligence and digital twin in internet of vehicles. IEEE Internet Things J, https://doi.org/10.1109/JIOT.2024.3517342
31. Asad MH, Khaliq S, Yousaf MH, Ullah MO, Ahmad A. Pothole detection using deep learning: A Real-Time and AI-on-the-Edge perspective. Adv Civ Eng. 2022;2022. https://doi.org/10.1155/2022/9221211.
32. Bibi R, et al. Edge AI-Based automated detection and classification of road anomalies in VANET using deep learning. Comput Intell Neurosci. 2021;2021. https://doi.org/10.1155/2021/6262194.
33. Alqahtani DK, Cheema A, Toosi AN. Benchmarking Deep Learning Models for Object Detection on Edge Computing Devices, 2024, [Online]. Available: http://arxiv.org/abs/2409.16808
34. Liu S, Zha J, Sun J, Li Z, Wang G. EdgeYOLO: an Edge-Real-Time object detector. Chin Control Conf CCC. 2023;2023–July:7507–12. https://doi.org/10.23919/CCC58697.2023.10239786.
35. Mittal P. A comprehensive survey of deep learning-based lightweight object detection models for edge devices. No 9 Springer Neth. 2024;57. https://doi.org/10.1007/s10462-024-10877-1.
36. Lin TY, et al. Microsoft COCO: common objects in context. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). May 2014;8693:740–55. https://doi.org/10.1007/978-3-319-10602-1_48. LNCS, no. PART 5.
37. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. The pascal visual object classes (VOC) challenge, *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010, https://doi.org/10.1007/S11263-009-0275-4/METRICS
38. Stacker L, et al. Deployment of deep neural networks for object detection on edge AI devices with runtime optimization. Proc IEEE Int Conf Comput Vis. 2021;2021–Octob:1015–22. https://doi.org/10.1109/ICCVW54120.2021.00118.
39. Amanatidis P, Karampatzakis D, Iosifidis G, Lagkas T, Nikitas A. Cooperative task execution for object detection in edge computing: an internet of things application. Appl Sci. 2023;13(8). https://doi.org/10.3390/app13084982.
40. Ganesh P, Chen Y, Yang Y, Chen D, Winslett M. YOLO-ReT: towards high accuracy Real-time object detection on edge GPUs. Proc - 2022 IEEE/CVF Winter Conf Appl Comput Vis WACV 2022. 2022;1311–21. https://doi.org/10.1109/WACV51458.2022.00138.
41. Liang S, et al. Edge YOLO: Real-Time intelligent object detection system based on Edge-Cloud Cooperation in autonomous vehicles. IEEE Trans Intell Transp Syst. 2022;23(12):25345–60. https://doi.org/10.1109/TITS.2022.3158253.
42. Kamath V, Renuka A. Deep learning based object detection for resource constrained devices: systematic review, future trends and challenges ahead. Neurocomputing. 2023;531:34–60. https://doi.org/10.1016/j.neucom.2023.02.006.
43. Yi A, Anantrasirichai N. A comprehensive study of object tracking in Low-Light environments. Sensors. 2024;24(13). https://doi.org/10.3390/s24134359.
44. Li J, Wang X, Chang Q, Wang Y, Chen H. Research on Low-Light environment object detection algorithm based on YOLO_GD. Electron. 2024;13(17). https://doi.org/10.3390/electronics13173527.
45. Royer A, Lampert CH. Localizing grouped instances for efficient detection in low-resource scenarios, *Proc. – 2020 IEEE Winter Conf. Appl. Comput. Vision, WACV* 2020, no. i, pp. 1716–1725, 2020. https://doi.org/10.1109/WACV45572.2020.9093288

46. Liu M, Luo S, Han K, Yuan B, DeMara RF, Bai Y. An Efficient Real-Time Object Detection Framework on Resource-Constricted Hardware Devices via Software and Hardware Co-design, 2024, [Online]. Available: http://arxiv.org/abs/2408.01534
47. Wahab F, Ullah I, Shah A, Khan RA, Choi A, Anwar MS. Design and implementation of real-time object detection system based on single-shoot detector and OpenCV. Front Psychol. 2022;13:1–17. https://doi.org/10.3389/fpsyg.2022.1039645. no. November.
48. Geiger A, Lenz P, Stiller C, Urtasun R. Vision meets robotics: The KITTI dataset. Int J Rob Res no Oct. 2013;32:1–6.
49. Lago A, Patel S, Singh A. Low-cost real-time aerial object detection and GPS location tracking pipeline. ISPRS Open J Photogramm Remote Sens. 2024;13:100069. https://doi.org/10.1016/j.ophoto.2024.100069. June.
50. Ye Y, Ma X, Zhou X, Bao G, Wan W, Cai S. Dynamic and Real-Time object detection based on deep learning for home service robots. Sensors. 2023;23(23). https://doi.org/10.3390/s23239482.
51. Liu Z, Chen C, Huang Z, Chang YC, Liu L, Pei Q. A Low-Cost and lightweight Real-Time Object-Detection method based on UAV remote sensing in transportation systems. Remote Sens. 2024;16(19). https://doi.org/10.3390/rs16193712.
52. Aamir SM, Malak K, Ali A, Aaqib M. Real Time Object Detection in Occluded Environment with Background Cluttering Effects Using Deep Learning, *7th Int. Work. Adv. Comput. Intell. Intell. Informatics*, p. 18, 2021, [Online]. Available: https://www.researchgate.net/publication/357434844
53. Alif MAR, Hussain M. YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain, pp. 1–31, 2024, [Online]. Available: http://arxiv.org/abs/2406.10139
54. Lee J, il Hwang K. YOLO with adaptive frame control for real-time object detection applications. Multimed Tools Appl. 2022;81(25):36375–96. https://doi.org/10.1007/s11042-021-11480-0.
55. Bello-Salau H, Onumanyi AJ, Adebiyi RF, Adekale AD, Bello-Salahuddeen R, Ajayi OO. A Critical Appraisal of Various Implementation Approaches for Real-time Pothole Anomaly Detection: Toward Safer Roads in Developing Nations †, *Eng. Proc.*, 56(1): 1–9, 2023, https://doi.org/10.3390/ASEC2023-15519
56. Smith TM. The Impact of Highway Infrastructure on Economic Performance, 1994. [Online]. Available: https://highways.dot.gov/public-roads/spring-1994/impact-highway-infrastructure-economic-performance
57. Berg C, Deichmann U, Selod H. How roads support development, 2015. [Online]. Available: https://blogs.worldbank.org/en/developmenttalk/how-roads-support-development
58. OECD. Impact Transp Infrastruct Invest Reg Dev. 2002. https://doi.org/10.1787/9789264193529-en.

Publisher's note